

Verifying the Secure Setup of UNIX Client/Servers and Detection of Network Intrusion

R. Feingold

This paper was prepared for submittal to the
"Photonics-East" International Society for Optical Engineers
Philadelphia, PA
October 22-26, 1995

July 1995



Lawrence
Livermore
National
Laboratory

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Verifying the secure setup of unix client/servers and detection of network intrusion¹

Richard Feingold
Secure Systems Services
Computer Security Technology Center
Lawrence Livermore National Laboratory
P.O. Box 808 L-303
Livermore, California 94551
Tel: (510) 422-1783 FAX (510) 423-8002
E-Mail: feingoldra@llnl.gov

Harry R. Bruestle, Tony Bartoletti, Allyn Saroyan, John Fisher
Computer Security Technology Center
Lawrence Livermore National Laboratory
P.O. Box 808 L-303
Livermore, California 94551
Tel: (510) 423-6224 FAX (510) 423-8002
E-Mail: cstc@llnl.gov

1. ABSTRACT

This paper describes our technical approach to developing and delivering Unix host- and network-based security products to meet the increasing challenges in information security. Today's global "Infosphere" presents us with a networked environment that knows no geographical, national, or temporal boundaries, and no ownership, laws, or identity cards. This seamless aggregation of computers, networks, databases, applications, and the like store, transmit, and process information. This information is now recognized as an asset to governments, corporations, and individuals alike. This information must be protected from misuse.

The Security Profile Inspector (SPI) performs static analyses of Unix-based clients and servers to check on their security configuration. SPI's broad range of security tests and flexible usage options support the needs of novice and expert system administrators alike. SPI's use within the Department of Energy and Department of Defense has resulted in more secure systems, less vulnerable to hostile intentions.

Host-based information protection techniques and tools must also be supported by network-based capabilities. Our experience shows that a weak link in a network of clients and servers presents itself sooner or later, and can be more readily identified by dynamic intrusion detection techniques and tools. The Network Intrusion Detector (NID) is one such tool. NID is designed to monitor and analyze activity on an Ethernet broadcast Local Area Network segment and produce transcripts of suspicious user connections. NID's retrospective and real-time modes have proven invaluable to security officers faced with ongoing attacks to their systems and networks.

Keywords: computers, networks, intrusion detection, security, Unix, tcp/ip, LAN, monitor

2. INTRODUCTION

The effective protection of information resources requires ongoing analysis of our computer and network resources. Retrospectively, on a continuing basis, we need to determine the security posture of our workstations. In real time we need to discover and respond to attacks before they cause significant damage.

¹This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

Most operating systems and applications are “practically,” rather than theoretically secure. That is, we create a protection strategy by:

- examining all of today’s threats, vulnerabilities, and countermeasures;
- analyzing host security profiles—file protections, password strengths, network services, patch levels, program integrity, configurations, etc.;
- and reasonably determining that (currently) there are no known vulnerabilities and, as far as we know, all countermeasures are in place.

This is a retrospective assessment: the profile might change due to lack of awareness, error, component failure, and so on; or some dedicated intruder might suddenly discover a new exploitation.²

Just as even the strongest door will not withstand a continual, unchecked assault, neither will most operating systems and applications.³ Further, the realities of most of today’s Local Area Networks (LANs) is that they are usually only as strong as their weakest host. Prudent security administration requires quickly determining when one’s resources are under attack in order to mitigate the potential adverse effects. For example, if we monitor someone copying an (unshadowed) Unix password file from host xyz, we:

- know that an account on xyz has been compromised (or worse),
- know to where the password file is being copied,
- probably know from where the attack is coming,
- can react immediately by changing all the passwords,
- can immediately decide to take xyz off the network,

and so on.

We use retrospective inspectors (e.g., SPI) and real time monitors (e.g., NID) to achieve the best of both worlds.

3. SPI OVERVIEW

The Security Profile Inspector (SPI) is a retrospective tool. It has been under continual development and enhancement for several years at the Computer Security Technology Center (CSTC) at Lawrence Livermore National Laboratory (LLNL). The CSTC is expanding SPI to support a distributed implementation and is extending its functionality to the desktop environment—Novell, Windows, and MS/PC-DOS. SPI is used extensively throughout the Department of Energy (DOE) and Department of Defense (DoD).

SPI assesses the security of Unix and VMS systems, reporting configuration vulnerabilities, bad passwords, and system file integrity violations. SPI:

- assesses and aids in establishing system security,
- measures system qualities and settings against recommended security standards,
- assists in incident detection, damage assessment, recovery, and preservation of evidence,
- provides a user-friendly interface, allowing for quick customization,
- provides on-line context-sensitive help, parameter management, automatic job scheduling, and report management.

Its powerful capabilities include:

²Some intruders literally read operating system and application source code looking for new vulnerabilities. Once discovered, they frequently post them on the Internet.

³Examples include password files and network file services. Given unlimited time and resources, passwords can be guessed; the same holds for guessing the correct “handle” for some network file services.

- Quick System Profile – configuration vulnerabilities.
- Access Control Test – access dependencies.
- Binary Authentication Tool – system binary patches.
- Password Security Inspector – password checks.
- Change Detector Tool – attribute/content deltas.
- Promiscuous Mode Checker – “sniffer” alert.
- Configuration Query Language – customize new checks.

SPI’s easy to use menus and time-saving features make it the perfect tool for busy system administration and security staff. Ideally, one runs SPI against a newly configured workstation, receiving an initial report and establishing a baseline. It is then best run periodically—the frequency determined by a formal or informal risk analysis.

4. NID OVERVIEW

The Network Intrusion Detector (NID) is a monitoring tool. It has been under continual development and enhancement for several years at the CSTC. The latter is expanding NID to support a real time monitor mode and add additional extensibility to its “attack signature” data base. NID is used extensively throughout the DOE and DoD.

NID protects networks by detecting unauthorized or malicious use. It helps identify unauthorized individuals and unauthorized or suspicious activities. NID:

- gathers network traffic, capturing only the specific portion of the network traffic that is vulnerable to attack, thus limiting the volume of data your security staff must analyze;
- detects and analyzes network traffic connections and scores them according to the likelihood of intrusion or unauthorized activity;
- summarizes in report form, to level of detail specified by user;
- generates a detailed transcript for review by the security staff;
- provides powerful, flexible “playback” capabilities.

NID provides critical dynamic intrusion detection capability that augments retrospective host-based security assessment tools. Figure 1 graphically shows the interactions of the capabilities.

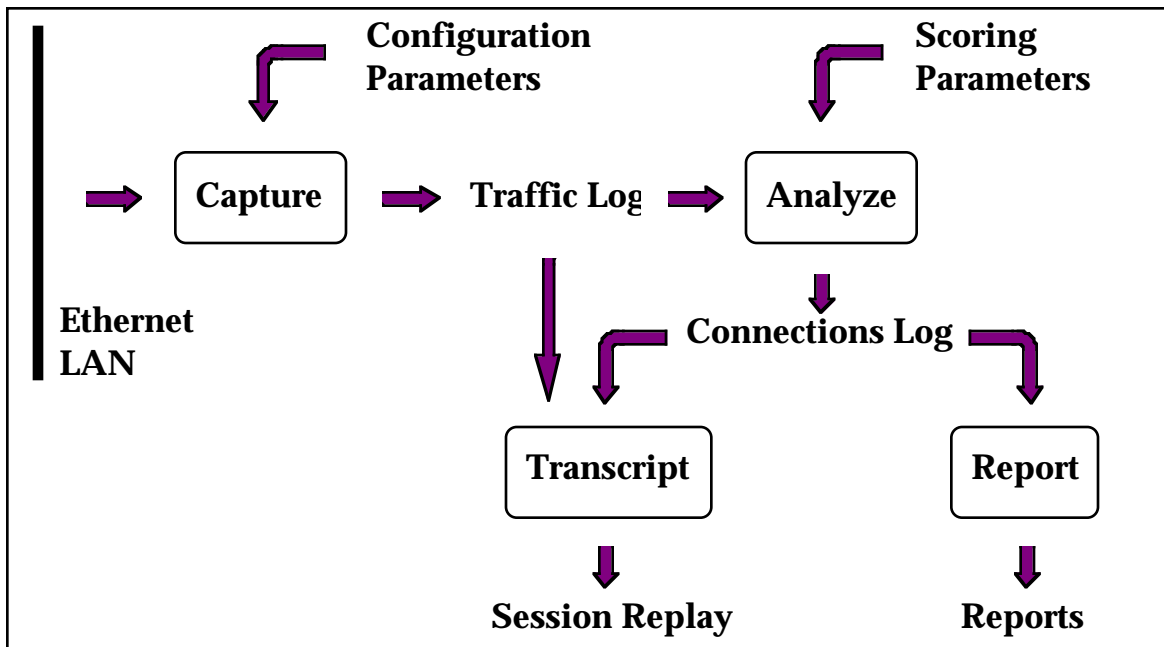


Figure 1. NID capabilities.

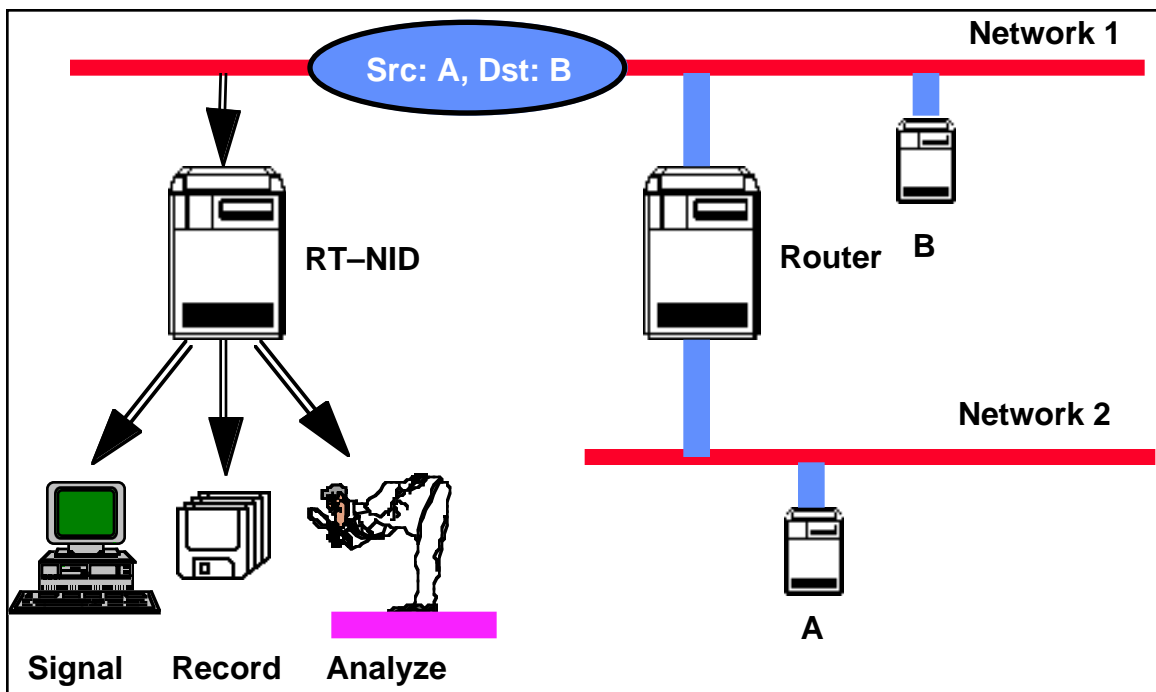


Figure 2. Normal NID configuration. This configuration is used for all situations except for detecting the IP spoofing attack. In the pictured configuration, the NID host and the hosts it is to protect are on the same network.

NID sits passively on a LAN, preferably on a highly secured, dedicated workstation.⁴ It is best run continually. Keep in mind that it can use a great deal of disk storage, so it is best to limit collection to significant source/destination nodes/domains and protocols of interest. NID is also a powerful assurance tool. In particular, by placing NID on both sides of a “Firewall,” or on the outside of an encrypted link,

⁴In particular, all unnecessary utilities, services, and accounts are removed.

NID can determine if the devices are functioning as intended from a security perspective.⁵

5. THE IMPACT OF SPI AND NID

SPI and NID are enabling technologies that empower computer and network administrators to effectively maintain parity with potential intruders from both inside and outside their LANs. They are living products, actively and currently maintained, with a stable source of continuing support. Furthermore, the CSTC environment, with its internationally recognized computer and network incident response team—CIAC⁶—provides current, realistic, and relevant input and support to the SPI and NID efforts. In particular, CIAC reports the latest Unix vulnerabilities being exploited and the latest attack methodologies for ultimate incorporation into the SPI and NID products. Thus, these products provide actual, “real world” defenses.

Cost effective security requires not only the best tools, but relevant and realistic policies and procedures. It is particularly frustrating to run SPI, discover vulnerabilities, and then be administratively unable to improve the situation. Likewise, it is counterproductive to discover an attack with NID and yet be unable to require workstation owners to make the necessary security improvements to thwart the attack.

While SPI and NID are effective products, they are not the only ones. They should be part of a continually re-evaluated suite of products chosen specifically for your environment. SATAN and/or ISS will provide a vulnerability perspective from the intruder or insider’s point of view. Unix accounting or other auditing tools may provide hints of compromise from a resource utilization perspective. Furthermore, SPI and NID must be maintained and used correctly and effectively. In the rapidly evolving world of ever more sophisticated attacks against your information resources, current updates to SPI and NID are essential. Furthermore, there needs to be ongoing training and awareness. Finally, your organization must assure the continued use and review of the reports generated by the products, specifying who executes the products and reviews the results and how frequently that is accomplished.

6. SPI AND NID AVAILABILITY

The products are available without charge to the sponsors and their contractors—currently DOE and DoD. For others, arrangements can be made to use the products, usually as part of a consulting effort.⁷

7. SPI TECHNICAL DETAILS⁸

7.1 Components.

Configuration Query Language (CQL) This is a high level, interpretive scripting language for extracting important system information, and presenting that information in a manner used by all other tools. CQL allows for complex queries to be made about files, users, and groups. Functionality that is not provided by the language may be introduced through C programming language functions.

Quick System Profile (QSP) This is an extensive CQL script, unique for each operating system, which looks for known vulnerabilities and common security problems. This tool is in a constant state of update, addressing the latest known vulnerabilities. The QSP script includes tests for

⁵For example, by observing what transactions get through the firewall, one can validate its configuration. One also might verify that downstream traffic from a link encryptor does not contain sensitive plaintext.

⁶The Computer Incident Advisory Capability for the Department of Energy.

⁷Contact the principal author for details.

⁸This section is extracted with editorial changes from “Security Profile Inspector (SPI) The Next Generation,” by Tony Bartoletti and John Fisher.

problems specific to a particular operating system.

Password Security Inspector (PSI) This tool is designed to uncover poorly chosen passwords. It attempts to match the user's encrypted password with common variations of the user's personal information, and words from selected custom dictionaries. An internal password inspection database is employed to allow the user to test only those passwords changed since the last time PSI was run, and allow the user to implement a discretionary password-aging policy.

Change Detector Tool (CDT) When initialized, this tool creates a database "snapshot" of important user, group, and file information. Users may define various subsets of files and accounts, and specify the attributes suitable for change detection reporting. On subsequent executions, CDT reports changes in this information relative to the snapshot, including when files, users, and groups have been changed, added, or deleted. The system administrator may then verify that the changes made were indeed intended. CDT is used to check for unauthorized additions of user accounts, to track group memberships, and to catch changes to file access permissions, ownerships, access and modification times, etc. File content changes may be tracked as well using crypto-checksums.

Access Control Test (ACT) This is a rule-based, goal-seeking system designed to assess sequential dependencies in computer access control mechanisms. Loosely based upon Bob Baldwin's "Kuang" tool⁹, this utility applies an external rule-base particular to Unix access controls.

Binary Authentication Tool (BAT) This tool ensures that all executables and libraries that make up the operating system are up-to-date (incorporate the latest patches) and authentic (are not Trojan Horses). BAT examines host file systems to identify known vulnerable system binaries and suggests the best patch or replacement binary to correct the problem. Authentication and patch information tables are provided by the SPI development team.

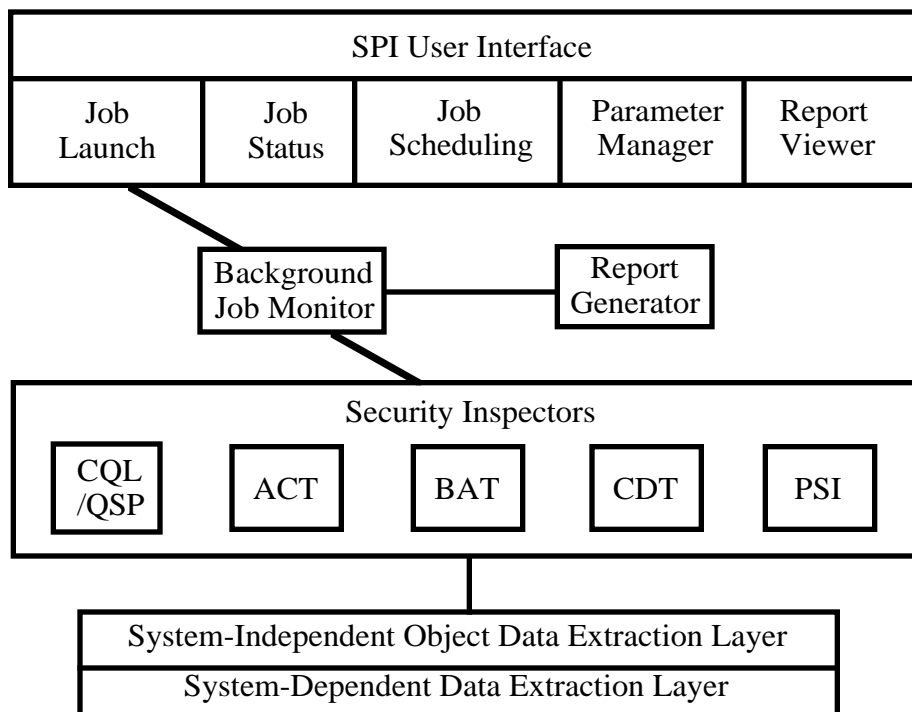
⁹Robert W. Baldwin, "Rule Based Analysis of Computer Security", MIT June 1987.

7.2 Ease of use.

All the SPI tools may be managed through a full-screen text-based user interface. Reasonable default values are provided for each inspection tool, and they can be scheduled to run at particular times. Every option and data field comes with on-line help specific to the current screen or data field selected, to aid novice users. For more experienced users, who wish to bypass the user interface, the command line options for all tools are fully documented through “man” pages.

All the SPI security inspection tools produce a machine-readable Common Output Report Format (CORF). The user interface employs the SPI report generator (RG) to convert this intermediate form, producing user friendly, informative reports. Direct access to the report generator and the raw tool output is provided to the advanced SPI user. The report generator uses customizable configuration files, allowing the end user to modify existing report formats, or create new reports. By using the common output format, results from multiple hosts or multiple inspection tools may be combined and used as input to additional forms of analysis or to produce more comprehensive global reports.

7.3 Current product structure.



7.4 Code architecture.

SPI uses multiple abstraction layers, to aid portability and adaptability. At the lowest level is the operating system, which provides its own set of primitives for accessing system information. All operating system dependent code is isolated in identifiable libraries, with a well-designed object oriented interface for extracting needed information. This provides a consistent system interface for all the inspection management tools.

In this programming interface, user, group, and file objects are constructed. Each object allows for consistent access to system data, independent of the operating system. To port all tools to a new

operating system, one need simply to port the object library. This design helped resolve portability issues. The recent porting of SPI to VMS demonstrated that SPI could be ported quickly to a non-POSIX environment, largely by rewriting the system-dependent data-extraction layer.

Providing a C programming language interface to system information, however, is not sufficient to encourage end users to expand SPI's security analysis capability.

The Configuration Query Language allows for high-level queries friendlier to users. It provides easy access to system information without having to compile or understand a cryptic programming language. CQL is quite powerful, enough so that the Quick System Profile tool is actually a collection of CQL scripts. Many of the other tools use CQL scripts internally. Site-specific system queries can be written easily by the end user.

One important advantage of CQL is its support for C programming language extensions. If a particular check can not be done through CQL alone, a C function may be written and called from a CQL script.

7.5 Design.

The CORF format serves as the communication mechanism between all SPI tools. It serves as a powerful and flexible means of storing both data and results. The SPI tools use CQL scripts (which generate CORF output) for extracting system information. CORF also serves as the database format, storing user, group, and file information. All the SPI inspection tools use CORF for reporting inspection results, including warnings, advisories, headings and summary information. Output from several tools may be combined and used to create new reports. Tools may be written which analyze output from several inspection tools, collected from a network of host machines, to produce results unobtainable from single host analysis.

Each tool is designed to be accessible by a wide range of users. Most users will feel comfortable modifying each tool's behavior through the user interface. For those who wish to have more direct access to each tool's functionality, each tool may be run from the command line. Standard Unix 'man' pages are provided which document command line parameters.

In short, we believe the current single-host SPI security assessment product has approached a zenith in fundamental security inspection and inspection management functionality. However, the effective use of security management personnel demands that network-wide inspections be easily conducted from a single command post, with sets of commands that automatically aggregate security inspections and report generation across large collections of host machines. This distributed inspection capability brings forth unique security management issues that the next generation SPI must address.

7.6 Future SPI efforts—Distributed Security Inspections.

The next-generation SPI is being designed to satisfy the following broad requirements:

- Automated and command-driven inspection of 50 or more remote host machines from a central command host, with consideration for upward scalability.
- Flexible inspection parameterization.
- Flexible job scheduling and automated job control.
- Flexible report aggregation capabilities.
- Robust performance in a variable environment.
- Host-to-host authentication, integrity assurance, and privacy assurance for all

command and data traffic.

8. NID TECHNICAL DETAILS¹⁰

8.1 Components

NID is a collection of tools that implement the following functions: network traffic collection, analysis of collected traffic, report generation, generation of a host-to-host transaction script, and replaying the script.

8.2 Monitoring modes

¹⁰This section is extracted with editorial changes from a presentation by Allyn Saroyan.

8.2 Monitoring modes.

	Traffic	Threat	Domain	Suspect	Server	Header
Security Domain	Crosses	Crosses	Inside	Outside	Dest in Src out	All
Service Used	Selected	Selected	All	All	Selected or All	All
Data Saved	Packets	Packets or None	Packets or None	Statistics	Statistics	Headers
Live Actions	–	Signal Context	Signal	Signal	–	–

Figure 3. NID monitoring modes.

Data can be collected in the modes shown in Figure 3. The names of the modes appear at the top of the figure and their major defining characteristics are shown on the left. The security domain is the sub-network you wish to protect. It generally consists of a set of host addresses. Depending upon the data collection mode, traffic will be captured if it crosses the security boundary, is entirely within the boundary, or is outside it. Services are application protocols such as ftp and telnet. Generally, one observes only those services that transmit ASCII data rather than binary. Most data collection modes allow the services to be specified. When packets are captured by a data collection mode, those packets may be retrospectively reviewed in detail. For modes that collect statistical data, a simple report generator and a binary file are available for use.

The “Traffic” mode is the traditional data capture mode in which packet data are captured if the source and destination of the packet cross your security domain boundary and use services you specify.

The “Threat” mode captures the same data as the traffic mode but also performs real-time analysis of that packet data and displays a signal if the cumulative threat value is above a threshold you select.

The “Domain” mode collects data only when the source and destination of a packet are within the security domain you specify. All services are captured. This mode may be used for both watching suspicious activity within a security domain and to detect IP spoofing when the NID host is located outside the security domain.

The “Suspect” mode collects statistics about network traffic whenever the source or destination of a packet is outside the security domain. All services are monitored. When a suspect is detected, a signal is given. The purpose of this mode is to alert you of unexpected use of your network. For example, you will be signalled if a new node is added to your network.

The “Server” mode collects statistics about network traffic whenever the source of a packet is outside your security domain and the destination is inside. Selected or all services may be monitored. This mode is used whenever you desire to know what services are being provided to hosts outside your domain.

The “Header” mode collects all packet headers on your network. This mode is used to collect data for statistical analysis of traffic loads.

8.3 Operations.

- **Intrusion detection steps:**
 - **Data collection** (capturing packets)
 - **Data analysis** (looking for signatures)
 - **Threat evaluation** (viewing the signatures in context)
- **Two operating modes:**
 - **Retrospective** (Three tools to perform the three steps)
 - **Real-Time** (Data collection and analysis in one tool)

Figure 4. NID operation.

NID operation is summarized in Figure 4. There are three steps for using NID. First, data are collected by capturing selected packets or statistics about those packets. Second, the data are analyzed for either threat signatures or statistical indications of threats. Third the threat is evaluated by viewing the threat indication in the context of the connection from which it was obtained.

NID operates in two modes: retrospective and real-time. In the retrospective mode, the three steps are embodied in three or more tools. In real-time mode, the first two steps are combined in one tool.

8.4 Real time actions.

- **Capture each packet.**
- ***If the packet crosses the security domain and is one of the desired services:***
- **Write packet to disk.**
- **Look for signatures.**
- ***If a signature is found:***
- **Evaluate the threat.**
- ***If the threat is above our threshold:***
- **Signal a threat has been detected.**

Figure 5. Real-time NID actions.

The actions listed in Figure 5 are modified slightly for the “domain” mode. In that mode, as soon as a packet within the security boundary appears, a message is displayed on the terminal. The other actions are the same.

8.5 Development plans.

- **Integration of all NID tools via a graphical user interface.**
- **Reduction of false-positive threat detection.**
- **User installation of threat descriptions which are read by signature, timing, and statistical use threat engines.**
- **Use of parameter sets for rapid mode switching.**
- **Automatic mode switching.**

Figure 6. NID development plans.

NID development plans are summarized in Figure 6. The future plans for NID always include the addition of new threat signatures. Only a few of the many additional enhancements needed to keep NID at the forefront of network intrusion detection are listed here.

Currently, analysts are faced with several different user interfaces for tools that must be run in a particular order. We propose to unify the tools from the user's point of view through a graphical user interface.

NID currently reports a large number of false-positive threat indications. We propose to incrementally become more sensitive to the context of a threat pattern rather than its mere existence.

Currently, NID must be reprogrammed to detect new intrusion mechanisms. We propose to use a set of recognition engines that understand threats in a generic manner. In order to recognize a particular threat, these engines would read user-installed threat descriptions and would allow the user to specify a threat value and the kinds of packets in which a threat may be found.

Currently, analysts edit two files to change the security domain and services of interest. We propose to allow multiple sets of files that are treated as a single entity for switching domains and services.

Currently, to change the data collection mode, one must restart the data collection program. We propose dynamic mode switching, both by user interaction and internally triggered events. For example, if a suspect appears on your network, you may wish to dynamically switch from the suspect mode (which gathers statistics) to the real-time analysis mode (which analyzes packets) in order to detect possible misuse.

9. CONCLUSIONS

SPI and NID are high quality, leading-edge products that effectively compliment each other. They are living, continually updated, actively supported, and incorporate the knowledge, experience, and expertise of an internationally recognized incident response team. These products are best used as part of a well thought out security plan involving relevant policies and procedures that uses other state-of-the-art tools as well. SPI and NID are freely available to the sponsors' constituencies and can be made available to others through value-added consulting contracts.

Technical Information Department • Lawrence Livermore National Laboratory
University of California • Livermore, California 94551

